MACHINE LEARNING ASSIGNMENT 2:
GLOBULAR CLUSTER PROPETIES

EXECUTIVE SUMMARY

This assignment explores the usefulness of machine learning for sparse astrophysics data. This is a purposeful endeavour. If proven effective on small data sets, machine learning could provide greater understanding into data that is difficult and expensive to acquire. However, this case is unlikely. It would contradict the notion that to generate efficient machine learning models in supervised learning, a large data set is required. To explore this issue, this assignment analyses a dataset of 147 observations on globular cluster properties. Globular clusters are the oldest known stellar systems in the local universe, indicative of the formation of the Milky Way ≥ 10 billion years ago.[1] With this data, the utility of machine learning methods is evaluated based on their ability to accurately predict cluster luminosity, as well as their understanding of cluster properties (variables). Finally, it is concluded that machine learning has limited utility on sparse data. When using small datasets, machine learning performances vary from poor to sufficient – but never effective. Most significantly, when the machine learning models are trained on sparse data, their predictions are unreliable due to their inability to repeatedly comprehend relationships between variables in the data set.

1 INTRODUCTION

1.1 MOTIVATION

For this assignment I answer the question of whether machine learning models are reliable methods for small data sets. To achieve this, I evaluate different machine learning methods using the multivariate dataset "GlobClus_prop". This is a regression dataset representing the properties of globular clusters, with the aim of predicting the absolute magnitude (Mv) of globular clusters.[2] To find which machine learning approach is suitable for this data, I will compare several tree-based models and a neural network using the performance metric of mean squared error MSE.

1.2 DATA

The GlobClus_prop dataset consists of 19 features (or variables) describing the properties of globular clusters and it has a total of 147 rows (or observations). The data is originally sourced from the catalogue of R.F Webbink (1985), working with the University of Illinois and the Joint Institute for Laboratory Astrophysics. The dataset has columns:

Galacticlong Galactic longitude (degrees)
Galacticlat Galactic latitude (degrees)
Rsol Distance from Sun (kpc)
RGC Distance from Galactic Centre (kpc)
Metal Log metallicity with respect to solar metallicity
Mv Absolute magnitude / Cluster Luminosity
CoreRadius Core radius (pc)
TidalRadius Tidal radius (pc)
Conc Core concentration parameter
logT Log central relaxation timescale (yr)
logRho Log central density (M/pc−3)
S0 Central velocity dispersion (km s−1)
Vesc Central escape velocity (km s−1)
VHB Level of the horizontal branch (mag)

[1] Feigelson, Babu (2012), 410.
[2] Feigelson, Babu (2012), 410.

EBV Colour excess (mag)
BV Colour index (mag)
Ellipt Ellipticity
Vt Integrated V magnitude (mag)
CSB Central surface brightness (mag per square arcsec)

1.3 EXPLORATORY DATA ANAYLSIS

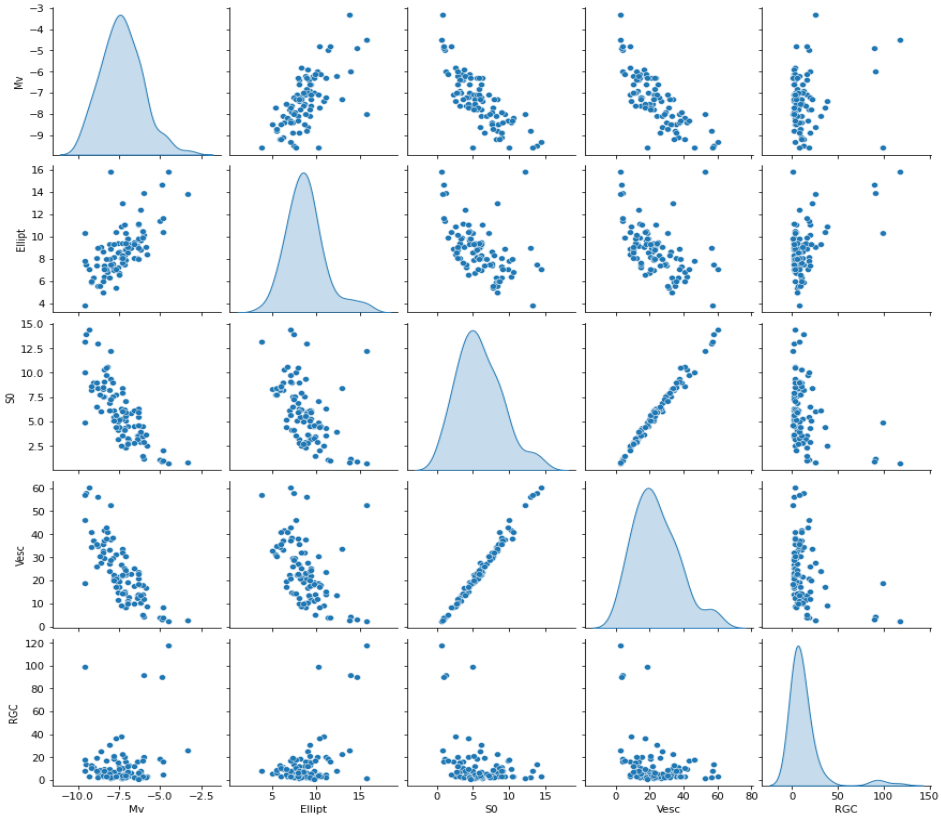
Before potential models are created, the dataset is analysed. Using Kendall's τ with the *cor* function is useful starting point to analyse multivariate relationships within the data. Here bivariate relationships to the response variable *Mv* – absolute magnitude or cluster luminosity are observed. (Table 1)

GalacticLong	**GalacticLat**	**Rsol**
-0.08	-0.07	-0.07
RGC	**Metal**	**CoreRadius**
-0.04	0.09	0.06
TidalRadius	**Conc**	**logT**
-0.29	-0.26	-0.06
logRho	**S0**	**Vesc**
-0.26	-0.65	-0.63
VHB	**EBV**	**BV**
0.02	0.09	0.13
Ellipt	**Vt**	**CSB**
0.47	-0.13	0.48

Table 1. The results of the correlation test.

In table 1, *Mv* shows a strong negative correlation to the dynamical variables *S0* – central velocity dispersion (km s−1), and *Vesc* – central escape velocity (km s−1). The photometric variable *CSB* – central surface brightness, and the structural variable *Ellipt* – ellipticity are also significant at a level $|\tau| > 0.40$. Importantly, the correlation coefficients indicate some variables are colinear. For instance, *GalacticLat* – galactic latitude, and *Rsol* – distance from the sun, both show a significance level of -0.07. Moreover, *S0* and *Vesc* are at significance levels -0.65 and -0.63 respectively. For this reason, collinearity must be analysed in further depth.

The shrinkage method lasso regression is optimal for this task as the data-generating mechanism is sparse. Of the 18 predictors in the data, the correlation coefficients indicate only 4 have a substantial effect on the response variable. Lasso regression is also beneficial as it performs best subset selection. The $\ell 1$ penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. In this enquiry, high values of lambda: $\lambda \geq 7$, lead to the inclusion of only *Vesc*. *Ellipt* and *TidalRadius* are selected at slightly lower values of lambda: $\lambda = 6$ and $\lambda \geq 4$. It is interesting that *TidalRadius* comes in third, despite having a reasonably small correlation with the response (-0.29). However, this may be expected since *TidalRadius* may be highly correlated to *Vesc*. In conclusion, *Vesc* and *Ellipt* are determined as highly significant variables.



x

Figure 1. A pairs plot of selected predictors and Mv, absolute magnitude. Linear relationships are observable, except or RGC—distance from galactic centre.

2 ANALYSIS AND MODELLING

2.1 GENERAL ALGEBRAIC MODELLING SYSTEM (GAMS)

The pairs plot shows both linear and nonlinear variables. For instance, the variable *RGC* – distance from galactic centre (kpc) shows clustering. (Figure 1) Therefore, to address both linear and nonlinear variables, this data will initially fit a GAM – General Algebraic Model. In this instance, the GAM predicts absolute magnitude using natural spline functions of *Ellipt* and *S0* at 5 and 3 degrees of freedom respectively. *Ellipt* and *S0* are used as lasso regression deemed them the two most significant variables. Thirdly, *RGC* is fitted as a qualitative predictor using a step function. *RGC* will be converted to the factor 'Subpopulation', reflecting whether a cluster is located at the halo or plane of a galaxy. If the cluster is located at distances beyond 80 kpc from the galactic centre, the cluster is classified as a halo subpopulation.[3] Clusters closer to the galactic centre are classified as plane subpopulations.[4]

The fitted GAM allows an extension of multiple linear regression and can be written as (1) [5]

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\ &= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i. \end{aligned} \tag{1}$$

[3] Alves-Brito et al. (2009), 1.
[4] Feigelson, Babu (2012), 410.
[5] James, (2021), 307.

The plots in figure 2 show the results of fitting the GAM. The natural spline for *Ellipt* shows the linear fit is clear. The plot shows holding *S0*, and subpopulation fixed, absolute magnitude tends to increase monotonically with greater ellipticity. This reflects the observation that most luminous globular clusters found in the Milky Way and Andromeda Galaxies are rounder than fainter clusters.[6] Thus, the model shows increased ellipticity decreases cluster luminosity.

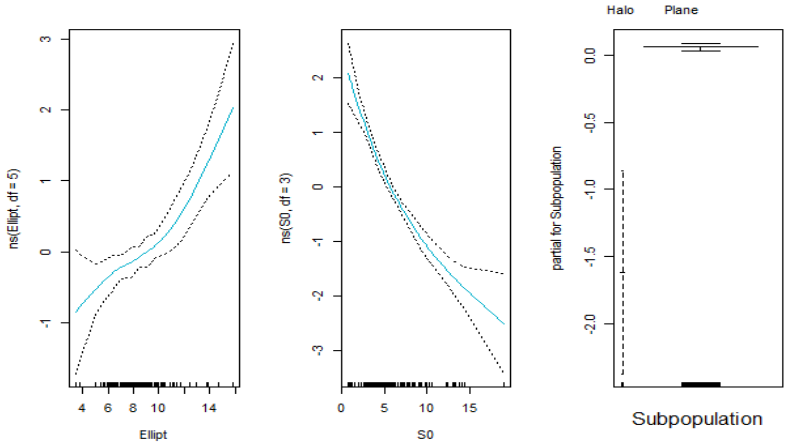


Figure 2. A GAM fitted on the full data set.

The linear fit of *Ellipt* is consistent, and most likely not overfitted. This contrasts with *S0*. The *S0* model may be overfitting at values around 5 km s−1 and predicting less accurately at values over 15 km s−1. The step function plot indicates the GAM recognises a difference in the absolute magnitude of halo clusters versus plane clusters. The predictions separate into two groups when *Mv* is predicted based on *RGC*. The plot indicates the model predicts clusters located on galactic planes to be less luminous than halo galaxies.

2.2 TREE BASED METHODS

2.2.1 DECISION TREE

Although the GAM makes clear predictions, it is important to understand how well machine learning methods comprehend connections between predictors and the response. For this reason, this assignment will now focus on tree-based methods. Beginning with a single decision tree, figure 3 illustrates how the predictor space is segmented into several samples before a prediction is made.[7] (Figure 3) In this tree, absolute magnitude is based on *S0*. The terminal nodes show if *S0* < 6.4 km s−1 then *Vesc*, *TidalRadius*, and *Ellipt* play a role in determining the luminosity of a cluster. This indicates the tree initially understands that *S0*, *Vesc* and *Ellipt* are crucial in predicting absolute magnitude. However, the model is confused by *TidalRadius* which is collinear with *Vesc*.

As pruning decision trees can improve performance, the tree was pruned and cross validated. However, the plotted cross-validation errors show in this case the unpruned or most complex tree performs best. (Figure 4) In keeping with the cross-validation results, the unpruned tree is used to make predictions on the test set. This yields a MSE of 0.61. The RMSE is approximately 0.78, indicating that this model leads to test predictions that are (on average) within approximately 0.78 Mv of the absolute magnitude of globular clusters. To gage the meaning of this, the luminosity of an object such as our sun is +4.83.[8] Highly luminous objects descend into minus values. For instance,

[6] Van den Bergh, (2008), 1.
[7] As the dataset is small, the data was split roughly in half to provide a training set (53 of 113 observations) and testing set with the remaining observations.
[8] Sun Fact Sheet. NASA. Website.

the Milky Way has a luminosity of −20.8.[9] Therefore, 0.78 is not very precise considering *Mv* is measured using low values.

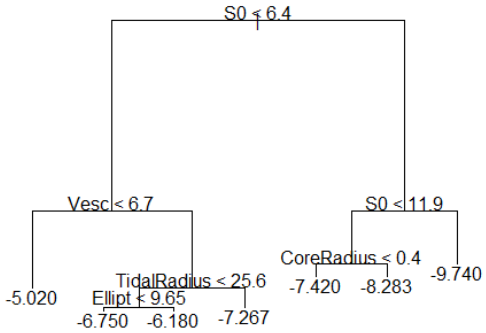


Figure 3. Single decision tree fitted on test data.

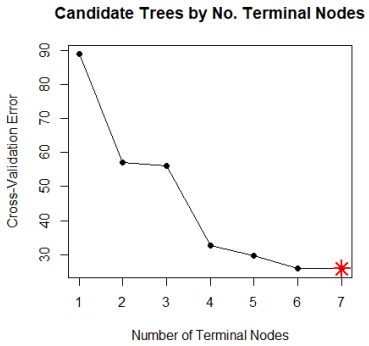


Figure 4. Plotted cross-validation errors.

2.2.2 RANDOM FORESTS AND BAGGING

Single trees suffer from high variance, meaning results can differ depending on the spilt sample. By contrast random forests offer a low variance alternative.[10] Random forests and bagging grow multiple trees by repeatedly taking samples from the training data set and combining them to yield a single consensus prediction.[11] This improves the predictive accuracy when compared to a single decision tree. [12] The method can be written as the following, where B is the number of trees in the forest (2)

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x). \qquad (2)$$

This assignment will implement two models on the training data and evaluate the MSE on the test data. In the first model, B = 500 trees and the in second B = 1000. Using two different values for B may highlight different performances based on the number of trees grown. Usefully, using a high value for B will not lead to overfitting.[13] Indeed, as the dataset is small, a higher number of trees may be beneficial in helping the machine learn meaningful connections between predictors. This is suggested by the lower MSE achieved by the model which fitted 1000 trees. (Table 2)

		MSE vs m Predictors	
Method	No. B Trees	m = 4	m = 18
Decision Tree	1	-	0.6065347
Forest	500	0.5239239	0.4920317
Forest	1000	0.5173915	0.4891065

Table 2. The MSE results from the decision tree, random forests, and bagging

[9] *Karachentsev, et al. (2004), 2056*
[10] James, (2021), 340.
[11] James, (2021), 327.
[12] James, (2021), 341.
[13] James, (2021), 341.

Typically, random forests are beneficial when dealing with colinear data sets, such as the globular cluster data. As random forests reduce the number of m predictors used in the model, prediction mistakes caused by collinear predictors are less likely.[14]

In random forests the number of m predictors is decided by taking the square root of the total predictors. For example, $m \approx \sqrt{18} = 4$.[15] Therefore, each time a split in a tree is considered, a random sample of 4 predictors are chosen as spilt candidates from the full set of 18 predictors.[16] This decorrelates the trees, leading to a thorough exploration of model space.[17] Despite this notion, results show reducing the number of predictors did not improve the MSE. (Table 2) Therefore, it can be concluded that bagged forests are more appropriate than random forests for small data sets. In other words, reducing m does not improve performance.

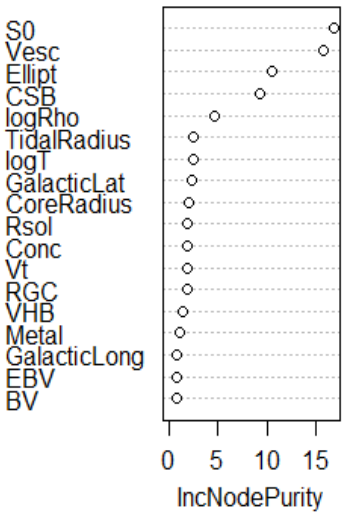


Figure 5. Variable importance results, showing node purity.

Nevertheless, the random forest approach is still useful. Random forest's exploration of model space evidently shows an intelligent understanding the relationship between predictors and absolute magnitude. (Figure 5). The plot shows the node purity of a random forest with 1000 trees. Purity represents how well each tree splits the data, the more impure the data set gets, the more splits are needed. As *S0, Vesc, Ellipt* and *CSB* are highly pure, the model successfully understands these predictors reduce splits.

2.2.3 BAYESIAN ADDITIVE REGRESSION TREES

So far, random forests and bagging have proven to perform better than single decision trees. However, the independent trees in random forests and bagging do not account for weighted predictors – predictors which may signalise higher performances. This is arguably a drawback. To combat this issue, the data is analysed using Bayesian additive regression trees (BART). Like random forests, BART uses the original data to make predictions from the average of regression trees. Yet, BART also takes into consideration a weighted sum of trees, each of which is constructed by fitting a tree to the residual of the current fit.[18] These trees can be denoted using K. BART will update each of the K trees upon each iteration. Upon every iteration, each response value is subtracted from all but the kth tree to find the partial residual. Consequently, BART can choose a perturbation to the tree from the previous iteration. BART favours perturbations which improve the fit to the partial derivative. This perturbation allows for a through exploration of model space. The output of BART is a collection of prediction models as follows (3) [19]

$$\hat{f}^b(x) = \sum_{k=1}^{K} \hat{f}_k^b(x), \text{ for } b = 1, 2, \ldots, B. \tag{3}$$

In this assignment different K values are tested to conclude how many trees are optimum for this data set. Additionally, degrees of freedom 3 and 5 will be fitted for the error variance.[20] Directly influencing the observation points, altering the degrees of freedom allows for extra exploration of

[14] James, (2021), 345.
[15] James, (2021), 343.
[16] James, (2021), 345.
[17] James, (2021), 352.
[18] James, (2021), 348.
[19] James, (2021), 350.
[20] Bayesian Additive Regression Trees. Rdocumentation. Website.

BART. It can be gaged how changes in freedom result in greater predictive accuracy or unexpected results. The degrees of freedom are specified by the sigdf argument.[21] After setting a seed to 1, the BART models ran on training data and their performance was evaluated using the MSE. Table 3 shows the results.

Model	No. K Trees	Degrees of Freedom	MSE
I	100	3	0.4587103
II	100	5	0.2259454
III	200	3	0.3370152
IV	200	5	0.348332
V	500	3	0.3891278
VI	500	5	0.470354

Table 3. Results from BART models

Overall, the MSE is notably improved using BART. This is an expected result given the greater robustness of the BART algorithm. The results show model II with K = 100 and 5 degrees of freedom has a significantly lower MSE than all other models. Therefore, model II could be considered the optimum model for this data set. Yet, the most reliable number of K trees is 200. This is because despite changes in the degrees of freedom, the model still yields an acceptably low MSE.

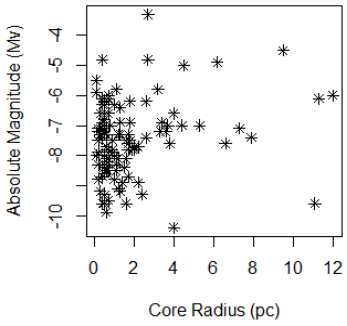


Figure 6. Scatter plot illustrating a weak relationship between absolute magnitude and core radius.

'Optimum' BART Model

CoreRadius	**Mv**	**EBV**
9.996	8.027	7.979
S0	**logRho**	**GalacticLong**
7.015	6.996	6.975
Rsol	**Vt**	**Conc**
6.766	6.066	6.041
Ellipt	**Metal**	**BV**
6.040	5.060	4.957
VHB	**Vesc**	**logT**
4.171	4.034	4.011
TidalRadius	**RGC**	**GalacticLat**
3.010	2.983	1.976

Table 4. Frequency of predictors using the BART model II

Although model II yielded the lowest MSE, it is problematic from a domain point of view. Considering how many times each predictor appears in the collection of trees, model II's predictions are most likely unreliable. (Table 4) Many predictors previously proven to highly correlate with absolute magnitude do not appear frequently. It should be expected that properties such as *S0, Vesc, Ellipt* and *CSB* appear most. By contrast, *CoreRadius* appears most frequently. Analysing a scatter plot of *Mv* and *CoreRadius* indicates there is no strong relationship between the two. (Figure 6) Therefore, model II may be making connections that are simply not there. It is highly likely model II would perform poorly on new data. Overall, it is likely the sparse data prevents BART from consistently making correct connections between predictors and absolute magnitude.

[21] Bayesian Additive Regression Trees. Rdocumentation. Website.

2.3 NEURAL NETWORKS

At this stage we can explore how neural networks perform on the data set in comparison to tree-based methods. Like the human brain, neural networks learn through pathways that are fired by an input. The issue in this assignment is the small input layer. The network may not be triggered to make intelligent predictions.[22] To explore this issue, it is important to first construct a neural network that likely has some capability performing on a small data set. Hence, the networks chosen for this assignment are as simple possible for a neural network. Layers are shallow, being either one or layers two deep. Moreover, the number of neurons or memory units in each layer is small – between 32 and 64. To aid the network in learning, the epochs will be set reasonably high. At 200 and 400 epochs the network should have enough time to comprehend the small amount of data.

The networks were built by importing *keras* and using the activate *relu* function. The optimiser chosen for this enquiry is Adam. This is because Adam yielded better learning rates and lower MSE than the pure stochastic gradient descent optimiser (SGD).[23] As this is a regression problem, the model's prediction accuracy or loss will be determined by MSE results.

		MSE Mv	
Network	Layer Architecture	200 Epochs	400 Epochs
I	{64} {32} {1}	0.533467	0.410384
II	{32} {32} {1}	0.875238	0.710225
III	{64} {64} {1}	0.948670	0.549105
IV	{32} {32} {32} {1}	0.926042	0.681736

Table 5. MSE results from neural networks

Analysing table 5, network I at 400 epochs is the best performing neural network with the lowest MSE. A plot of the loss history for network I shows a low to good learning rate during training.[24] The validation loss does not appear to dissimilar to the loss value and does not follow the loss function too closely. This suggests the model should make reasonably accurate predictions. (Figure 7) However, the performance of network I is yet to be determined by its predictions.

Comparing the loss history of network, I (figure 7) and III (figure 8), demonstrates adding a more neurons in the second hidden layer decreases the learning rate. This suggests simple neural networks are the best solution for small data sets. Additionally, for this data, the MSE error indicates the learning rate is better when the number of neurons is decreased by layer as opposed staying the same.

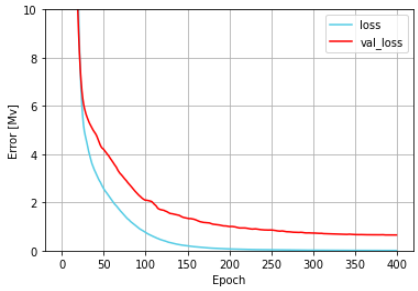


Figure 7. Plotted loss history of network I at 400 epochs.

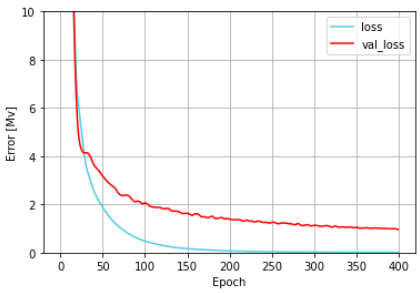


Figure 8. Plotted loss history of network III at 400 epochs.

[22] Efron, *et al*. (2016), 364.
[23] Tf.keras.optimizer.Adam. TensorFlow. Website.
[24] Convolutional Neural Networks for Image Recognition. GitHub. Website.

Analysing an array of 10 sample predictions on the test features shows the neural network makes realistic predictions for absolute magnitude. Here, the sample predictions range from approximately -10 to - 5 Mv. Similarly, the dataset records absolute magnitude ranges from -10.4 to -3.3. As the final predictions are realistic, it can be argued a simple neural network is a useful tool in analysing this small dataset.

On the other hand, the plot of test predictions shows the model struggles to make predictions for the true values around -9 to -11. (Figure 10) A similar issue is evident when network I is asked to predict outcomes of a regression problem between only *S0* and *Mv*. (Figure 11) Where there are exceptionally few observations, the network cannot learn a predictive relationship.[25] Indeed, the network may be ignoring these observations all together. This is a significant problem. Overall, even simple neural networks are unreliable in exploring small data sets.

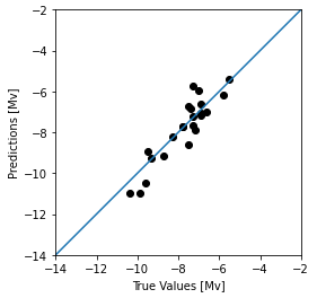


Figure 10. Plotted predictions of network I, accounting for multiple variables.

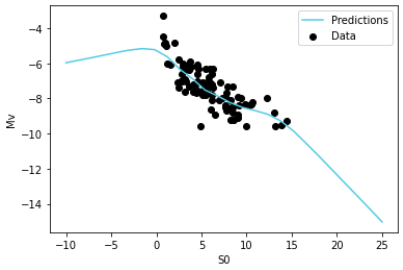


Figure 11. Plotted predictions of network I, accounting for *Mv* and *S0*.

3 DISCUSSIONS

3.1 FINDINGS

Lowest and highest MSE values for each method type

Model	Lowest MSE	Highest MSE
Decision Tree	0.6065347	0.6065347
Forest (Bagging)	0.4891065	0.4920317
Random Forest	0.5173915	0.5239239
BART	0.2259454	0.470354
Neural Network	0.410384	0.948670

Table 6. MSE results of the machine learning methods. Lowest MSE reflects the best performing models, while highest MSE reflects the worst performing models.

Overall, this assignment showed that supervised machine learning methods are not efficient tools for sparse data sets. In table 6 the BART method yields the lowest MSE of 0.22, indicating it is the best model for this data set. However, in section 2.2.3, results also showed BART poorly understood correlations between predictors and the response *Mv*. For this reason, BART is not a reliable method for sparse data. It must be noted that depending on the degrees of freedom and number of trees grown, BART's understanding of variable relationships changed. Therefore, with further experimentation it is possible BART's comprehension of variables can be improved.

Most notably neural networks behave unpredictably when the data set is small. For instance, MSE results for neural networks range from 0.41 to ≈ 0.95.(Table 6) Some predictions made by the network are realistic. However, prediction plots suggest the network is incapable of learning from

[25] Efron, *et al*. (2016), 363.

exceptionally sparse data. Thus, neural networks generally perform very poorly on the globular cluster data, yielding them unusable for small data sets.

By contrast, random and bagged forests are more consistent than neural networks. For example, both methods yield MSE results ≈ 50. (Table 6) Additionally, the difference between the best and worst performing models is slight, proving their performance is consistent. However, it must be noted that at 0.48, even the best performing bagged forest, does not have a sufficiently low MSE to argue this approach is useful.

The only potential utility in supervised machine learning for sparse data sets lies in the random forest's ability to comprehend variable relationships. The results of an importance plot indicate random forests intelligently understand the significance of the variables *S0, Vesc, Ellipt* and *CSB* to the response variable *Mv*. These variables were proven to have the highest correlation to *Mv* in initial data exploration. This suggests random forests are less likely to be confused by colinear variables. Consequently, these results indicate random forests are the best machine learners when the data set sparse.

3.2 FURTHER WORK

Given more time, I would further this work by examining how feeding different variable sets into the model affects predictive performance. Referring to initial data exploration, lasso regression proved only *S0, Vesc, Ellipt* and *CSB* are highly significant to *Mv*. Additionally, lasso regression indicated the variable *TidalRadius* is colinear with *Vesc*. Based on these findings, it would be useful to evaluate the MSE of each model if *TidalRadius* was removed. Notably, this may improve the performance of the single decision tree and the BART model, both of which mistakenly perceived *TidalRadius* as significant. Additionally, the neural network could be reassessed after being fed only the 4 most significant variables. In the best case, this exploration would help the neural network learns from all data points.

REFERENCES

Alves-Brito, A; Forbes, D; Mendel, J; Hau, G and Murphy, M (2002) 'The Outer Halo Globular Clusters of M31', Centre for Astrophysics and Supercomputing, Hawthorn

Efron, B. and Hastie, T. (2016). Computer Age Statistical Inference: Algorithms, Evidence, and Data Science. Institute of Mathematical Statistics Monographs. Cambridge: Cambridge University Press.

Feigelson, E and Babu, J. (2012). Modern Statistical Methods for Astronomy with R Applications, Cambridge

Grayzeck, E. (2018) 'Sun Fact Sheet' < https://nssdc.gsfc.nasa.gov/planetary/factsheet/sunfact.html > Last accessed: 20 February 2022]

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An Introduction to Statistical Learning: with Applications in R. Springer Texts in Statistics. New York: Springer.

Karachentsev, I; Karachentseva, V; Huchtmeier, W; Makarov, D. 'A CATALOG OF NEIGHBORING GALAXIES' The Astronomical Journal, 127:2031–2068 (2004)

R Documentation, 'Bayesian Additive Regression Trees' < https://search.r-project.org/CRAN/refmans/BayesTree/html/bart.html > [Last accessed: 20 February 2022]

TensorFlow, 'tf.keras.optimizers.Adam' < https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam> [Last accessed: 20 February 2022]

Van den Bergh, S. (2008). 'The Flattening of Globular Clusters', Dominion Astrophysical Observatory, Herzberg Institute of Astrophysics, Herzberg